

Rapport de modélisation V1

Pour ce projet nous allons nous baser sur la billetterie du cinéma Grand Rex et compléter par certaines fonctionnalités provenant de L'Accor Arena.

Un utilisateur est déterminé par son email, par son statut s'il est connecté ou non, vip etc il a une capacité de réservation différente. Chacun a son historique d'achat, remboursement, échange etc, et peut détenir un avantage.

Le cinéma propose un évènement, un évènement peut contenir des sous évènements tel que la rencontre avec les acteurs par exemple.

Dans le cas d'une séance de cinéma, les catégories se distinguent par classic ou vip en général mais pour une séance de concert, ça se distingue par « cat1, cat2, etc ».

L'échange de billet est permis dans la limite de chaque utilisateur.

Au moment de l'achat d'un billet, une pré-réservation est automatiquement mise en place le temps d'un délai fixe (15min pour l'instant).

Dans la situation d'une surcharge, un système de file d'attente sera mis en place, dans la base de données, un créneau avec une heure de début et de fin fixe d'avance.

(Les clés candidates, dépendances fonctionnelles non triviales et contraintes se trouvent sur la prochaine page).

Table	Clé Candidates (Naturelles)	DF non triviales
Utilisateur	email	email -> nom,id_avantage,id_statut
Avantage	nom,type,valeur	(nom,type,valeur) -> description,date_expiration
StatutUtilisateur	Nom	Nom -> limite_places
Evenement	Nom,date_debut	(nom,date_debut) -> description,date_fin
Seance	Id_evenement,date_heure	(id_evenement,date_heure) -> type
Categorie	Id_seance,nom	(id_seance,nom) -> prix
Billet	id_utilisateur, id_seance, id_categorie	(id_utilisateur, id_seance, id_categorie) -> prix_final, date_achat
PreReservation	id_utilisateur, id_seance, id_categorie	(id_utilisateur, id_seance, id_categorie) -> date_expiration, statut
CreneauConnexion	id_utilisateur, date_heure_debut	(id_utilisateur, date_heure_debut) -> date_heure_fin, statut
PolitiqueEtablissement	id_lieu, type_restriction	(id_lieu, type_restriction) -> valeur
EchangeBillet	Id_billet	id_billet → id_utilisateur_vendeur, id_utilisateur_acheteur, prix, statut
HistoriqueTransactions	id_utilisateur, date_action	(id_utilisateur, date_action) → action, details

Table	Attribut	Type de Contrainte	Motif
Utilisateur	Email	Unique	Chaque utilisateur a un email unique
Avantage	date_expiration	CHECK (date_expiration >= CURRENT_DATE)	Empêcher les avantages passées
Evenement	date_debut	CHECK (date_debut < date_fin)	Assurer la cohérence des dates
Seance	date_heure	CHECK (date_heure >= CURRENT_DATE)	Empêcher des séances passées
Categorie	Prix	CHECK (prix >= 0)	Empêcher les prix négatifs
Billet	prix_final	CHECK (prix_final >= 0)	Empêcher des valeurs négatives
Billet	date_achat	CHECK (date_achat <= date_heure)	Empêcher des achats dans le passé
PreReservation	Statut	CHECK (statut IN ('en_attente', 'confirme', 'expire'))	Valeurs valides uniquement
CreneauConnexion	Statut	CHECK (statut IN ('attribue', 'utilise', 'expire'))	Valeurs valides uniquement
PolitiqueEtablissement	type_restriction	CHECK (type_restriction IN ('quota', 'age', 'abonnement', 'autre'))	Valeurs valides uniquement
EchangeBillet	Statut	CHECK (statut IN ('disponible', 'vendu', 'annule'))	Valeurs valides uniquement
HistoriqueTransactions	date_action	CHECK (date_action >= CURRENT_DATE)	Assurer une date cohérente

Contraintes avec TRIGGER:

Billet : Empêcher l'achat d'un billet si la limite de places est atteinte.

PreReservation : Annuler automatiquement les pré-réservations expirées.

CreneauConnexion : Limiter un utilisateur à un seul créneau de connexion actif.

EchangeBillet : Empêcher un échange si l'acheteur possède déjà un billet pour la séance.